94-775/95-865 Lecture 12: Time Series Analysis With Recurrent Neural Nets

George Chen
RNNs

What we’ve seen so far are “feedforward” NNs
RNNs

What we’ve seen so far are “feedforward” NNs

What if we had a video?
RNNs

Feedforward NN’s: treat each video frame separately

Time 0

Time 1

Time 2

...
RNNs

Feedforward NN’s: treat each video frame separately

RNN’s: feed output at previous time step as input to RNN layer at current time step

In *keras*, different RNN options: *SimpleRNN*, *LSTM*, *GRU*

Recommendation: don’t use *SimpleRNN*
RNNs

Feedforward NN’s: treat each video frame separately

RNN’s: feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
- `SimpleRNN`
- `LSTM`
- `GRU`

Recommendation: don’t use `SimpleRNN`
Example: SimpleRNN

memory stored in `current_state` variable!

```python
current_state = 0
for input in input_sequence:
    output = activation(np.dot(input, W) + np.dot(current_state, U) + b)
    current_state = output
```

Activation function could, for instance, be ReLU

Parameters: weight matrices $W$ & $U$, and bias vector $b$

Key idea: it's like a dense layer in a for loop with some memory!
RNNs

Feedforward NN’s: treat each video frame separately

RNN’s: feed output at previous time step as input to RNN layer at current time step

In *keras*, different RNN options: SimpleRNN, LSTM, GRU

Recommendation: don’t use SimpleRNN

Time series

RNN layer

readily chains together with other neural net layers

like a dense layer that has memory
RNNs

Feedforward NN’s: treat each video frame separately

RNN’s: feed output at previous time step as input to RNN layer at current time step

RNN layer readily chains together with other neural net layers

like a dense layer that has memory

In *keras*, different RNN options: SimpleRNN, LSTM, GRU

Recommendation: don’t use SimpleRNN
RNNs

Feedforward NN’s:
treat each video frame separately

RNN’s:
feed output at previous time step as input to RNN layer at current time step

like a dense layer that has memory

readily chains together with other neural net layers

In keras, different RNN options:
SimpleRNN, LSTM, GRU

Recommendation:
don’t use SimpleRNN
**RNNs**

Example: Given text (e.g., movie review, Tweet), figure out whether it has positive or negative sentiment (binary classification)

Common first step for text: turn words into vector representations that are semantically meaningful.
(Flashback) Do Data Actually Live on Manifolds?

Example: Given text (e.g., movie review, Tweet), figure out whether it has positive or negative sentiment (binary classification).

Common first step for text: turn words into vector representations that are semantically meaningful.

In `Keras`, use the `Embedding` layer.

Classification with > 2 classes: dense layer, softmax activation.

Classification with 2 classes: dense layer with 2 neurons & softmax equivalent to dense layer with 1 neuron & sigmoid activation (called **logistic regression**).

For loss function, replace category cross entropy with **binary cross entropy**.
Word Embeddings

Example of self-supervised learning

*Even without labels*, we can set up a prediction task!

*Hide* part of training data and try to predict what you’ve hid!

I’ll talk more about self-supervised learning next lecture (it’s a clever application of predictive data analytics concepts)
RNNs

Demo
RNNs

• Neatly handles time series in which there is some sort of global structure, so memory helps.

  • If time series doesn’t have global structure, RNN performance might not be much better than 1D CNN.

• An RNN layer by itself doesn’t take advantage of image/text structure!

  • For images: combine with convolution layer(s)
  • For text: combine with embedding layer
A Little Bit More Detail

Time series → RNN layer → output prediction
SimpleRNN tends to forget things quickly.

\[
\text{output}[t] = \text{activation}(\text{np.dot(input}[t], W) + \text{np.dot(current}_\text{state}, U) + b)
\]
Add explicit long-term memory!

But need some way to update long-term memory!
Add explicit long-term memory!

But need some way to update long-term memory!
Add explicit long-term memory!

But need some way to update long-term memory!
But need some way to update long-term memory!

Add explicit long-term memory!

Called a “long short-term memory” (LSTM) RNN

Remembers things longer than SimpleRNN

Time $t - 1$

Long-term memory updater

Time $t$

output $t - 1$

output $t$